

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

THIS PAGE BLANK (USPTO)

PCTWORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06K 9/00	A1	(11) International Publication Number: WO 00/38098
		(43) International Publication Date: 29 June 2000 (29.06.00)

(21) International Application Number: PCT/US99/29944

(22) International Filing Date: 16 December 1999 (16.12.99)

(30) Priority Data:
09/218,975 22 December 1998 (22.12.98) US(71) Applicant (for all designated States except US): EXPAND
NETWORKS LTD. [IL/IL]; P.O. Box 58142, 61580 Tel
Aviv (IL).(71) Applicant (for TJ only): FRIEDMAN, Mark, M. [US/IL];
Alharizi Street 1, 43406 Raanana (IL).(71)(72) Applicant and Inventor: MARCO, Talmon [IL/IL]; Ha-
banim Street 39, 46379 Herzliya (IL).

(72) Inventor; and

(75) Inventor/Applicant (for US only): KALKSTEIN, Nir [IL/IL];
Hanassi Street 24, 46602 Herzliya (IL).(74) Common Representative: FRIEDMAN, Mark, M.; c/o Casto-
rina, Anthony, Suite 207, 2001 Jefferson Davis Highway,
Arlington, VA 22202 (US).(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG,
BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB,
GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG,
KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK,
MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG,
SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU,
ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL,
SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ,
MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE,
DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE),
OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML,
MR, NE, SN, TD, TG).**Published***With international search report.**Before the expiration of the time limit for amending the
claims and to be republished in the event of the receipt of
amendments.*

(54) Title: SYNTHETIC HISTORY FOR ADAPTIVE DATA COMPRESSION

(57) Abstract

Data is compressed (20) and decompressed (28) using synthetic (30) history data. The synthetic history data is used to provide preliminary history data to a data compressor and a data decompressor at the start of data compression and data decompression operations, respectively. Several synthetic histories may be defined, each of which is associated with a given data type. The data type of data to be compressed may be ascertained to determine which synthetic history is to be used for that data. The synthetic history data may consist of "typical text" that contains strings which are likely to appear in a document of a certain type. The "typical text" is compressed to provide the preliminary history data.

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

ANNEX 1

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Sénégal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

SYNTHETIC HISTORY FOR ADAPTIVE DATA COMPRESSION

FIELD OF THE INVENTION

5 The present invention relates to data compression and, more specifically, to compressing and decompressing data using pre-defined history-related information.

BACKGROUND OF THE INVENTION

10 Data compression algorithms convert data defined in a given format to another format so that the resulting format contains fewer data bits (i.e., the ones and zeros that define digital data) than the original format. Hence, the data is compressed into a smaller representation. When the original data is needed, the compressed data is decompressed using an algorithm that is
15 complementary to the compression algorithm.

 Data compression techniques are used in a variety of data processing and data networking applications. Personal computer operating systems use data compression techniques to reduce the size of data files stored in the hard disk drives of the computer. This enables the operating system to store
20 more files on a given disk drive. Data networking equipment use data compression techniques to reduce the amount of data sent over a data network. For example, when a web browser retrieves a file from a web server, the file may be sent over the Internet in a compressed format. This reduces the transmission time for sending the file and reduces the usage of
25 the network, thereby reducing the cost of transmission.

 Many compression schemes use translation data dictionaries that contain a series of mappings between the original data and the compressed representations of the actual data. For example, the letter "A" may be represented by the binary string "010." To ensure that data is decompressed
30 accurately, the compressor and decompressor use identical dictionaries. In this case, the dictionaries may be supplied to each component (compressor or decompressor) or dynamically created by each component using known algorithms.

A dictionary typically is derived from the data according to a selected scheme relating to various statistical information gathered therefrom, such as the frequencies of certain patterns in the data. For example, the length of the bit representation in the encoding table for each of the encoded data patterns may be selected so that it is inversely related to the frequency of occurrence of the corresponding patterns.

Hereinafter, the term "text" refers to a stream of data bits which is provided as a unit to the compression algorithm and includes, but is not limited to, word data from a document, image data and other types of data. As noted above, the text can have features or characteristics such as internal patterns of data.

There are several well-known data compression methods which may be classified according to how they generate and use dictionaries. Static compression algorithms use static dictionaries. That is, the algorithms do not affect, update or otherwise change the dictionary for a given unit of text.

Dynamic compression algorithms, on the other hand, constantly update or change the dictionary according to features or characteristics of the text based on a selected scheme. In semi-static compression algorithms, the dictionary is occasionally updated or changed according to the text based on a selected scheme. Hereinafter, the term "adaptive compression algorithm" refers to a dynamic or semi-static algorithm in which the history is either constantly or occasionally updated or changed according to data pattern variations encountered in the text.

In adaptive compression algorithms schemes, the dictionary is commonly referred to as a history. At any given moment in time during the compression process, the history is a representation of some or all of the data that has been processed by the compression algorithm. An example using the text "when and where is to be determined" (underlining added) is illustrative. When compressing data, the compression algorithm checks each string of data to determine whether that string already appears in the history. Thus, when the compression algorithm reaches the underlined "whe", the string "when" is already in the history. As a result, the compression algorithm

can reference (and, consequently, compress) the first the letters of "where".

From the above, it may be observed that the compression ratio for the text depends on the number of matches that are found in the history and on the length of the matched terms (given that a long term may be represented by a relatively short representation).

The data in the history may consist of ordinary text (as in the example above). Alternatively, the history may consist of more sophisticated representations such as hash data or a linked list data.

Adaptive algorithms have a number of advantages. For example, these algorithms permit the history be adjusted to best reflect the data patterns in the text. Thus, adaptive algorithms have, in essence, a "learning" capability. Furthermore, the history need not necessarily be transmitted along with the encoded data, but rather can be fully rebuilt at the receiving end from the encoded data during decompression. Thus, this class of techniques is particularly well suited for data compression in a communication system.

Examples of adaptive data compression techniques include the well-known Lempel-Ziv algorithms known, respectively, as LZ77 and LZ78, for constructing the encoding table (Ziv J., Lempel A.: A universal algorithm for sequential data compression, IEEE Transactions on Information Theory, Vol IT-23, (1977) pp. 337-343; Ziv J., Lempel A.: Compression of individual sequences via variable rate coding, IEEE Transactions on Information Theory, Vol IT-24, (1978) pp. 530-536).

Waterworth (Waterworth J.R.: Data compression system, US Patent No 4,701,745, October 20, 1987) and Whiting et al. (Whiting D.L., George G.A., Ivey G.E.: Data compression apparatus and method, US Patent No 5,016,009, May 14, 1991; Whiting D.L., George G.A., Ivey G.E.: Data compression apparatus and method, US Patent No 5,126,739, June 30, 1992) provide efficient implementations of the Lempel & Ziv LZ77 technique for identifying data patterns in the text. A similar fast implementation is given by Williams (Williams R.N., An extremely fast Ziv-Lempel data compression algorithm, Proceedings Data Compression Conference DCC '91, Snowbird, Utah, April 8-11, 1991, IEEE Computer Society Press, Los Alamitos, CA, pp.

362-371).

In addition, Brent (Brent R.P.: A linear algorithm for data compression, The Australian Computer Journal, Vol 19, (1987) pp. 64-68) provides a static technique that takes advantage of both LZ77 and the Huffman encoding scheme. The Huffman coding scheme is discussed, for example, in Huffman
5 D.: A method for the construction of minimum redundancy codes, Proceedings IRE, Vol 40, (1952) pp. 1098-1101.

Although these well-known data compression techniques have been successfully employed in many applications, there is an ever-present need for
10 improved compression techniques, particularly in data networking applications.

SUMMARY OF THE INVENTION

The adaptive compression scheme of the present invention provides
15 preliminary history data that is available when compression is commenced on a given text. This preliminary history data is generated from pre-defined history-related information referred to herein as synthetic history data. By providing preliminary history data, the method of the invention increases the probability that there will be an increased number of matches between the
20 text and the history file data during the compression process. In particular, more matches may occur for the portion of the data in the text that is processed during the early stages of the compression process.

This advantage of the invention may be better understood by comparison with conventional adaptive compression techniques that
25 commence compressing a given text with an empty history file. In the conventional case, there is never a match for the very first element of data in the text that is processed by the compression algorithm because there is no history data. Moreover, there typically will be few matches of the text and the history data until a relatively large amount of history data is added to the
30 history file. In contrast, the present invention provides more efficient compression because the text will be compressed to a higher degree due to the increased number of matches between the text and the history file data.

In one embodiment of the invention, the data for the synthetic history files is defined based on an expected correlation between a given text to be compressed and a predefined body of text is related in some manner to the text to be compressed. Given two texts of a similar type (e.g., articles written
5 in French), it is expected that there will be a certain correlation between words appearing in two or more of the these texts. It may be understood, then, that a wide variety of category types may be defined. For example, books that deal with environmental engineering, HTML documents, image data, etc.

In one embodiment, the synthetic history information consists of a
10 "typical text." The "typical text" is defined as a text of a pre-defined length that contains strings which are most likely to appear in a document of a certain type. For example, an HTML (hyper-text mark-up language) document will likely contains such strings as "<HTML>" and "IMG SRC=".

The compression algorithm processes the "typical text" to generate the
15 preliminary history data. When a target text of a given type (e.g., HTML) is to be compressed, the associated, pre-defined "typical text" is compressed before compressing the target text. This creates preliminary history data that is used to compress the target text. In this way, when the target text is compressed, there may be a high probability that certain strings that appear
20 in the target text are already present in the history and, therefore, may be referenced.

In one embodiment, the compressed data that results from the typical text is discarded. This eliminated the need, for example, to transmit this data to a remote equipment where the compressed target text is to be
25 decompressed.

When the compressed target text is to be decompressed, the decompression algorithm performs steps complementary to those described above. Specifically, the decompression algorithm first decompresses a compressed "typical text" thereby creating a preliminary history. Next, the
30 decompression algorithm decompresses the compressed target text using the preliminary history. The decompressed text that results from the compressed "target text" is then discarded. Thus, the decompression stage produces an

accurate replica of the original target text.

In one embodiment, the invention is implemented in a pair of devices installed in a data network such as the Internet. For example, the devices may be installed between a pair of routers that define an IP hop. The device
5 on the sending end of the hop intercepts each packet that the router sends over the hop and determines whether that packet contains a type of text that may be compressed using synthetic history.

The device on the other end of the hop also intercepts each incoming packet and determines whether that packet was compressed using synthetic
10 history. If so, the packet is decompressed using synthetic history.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features of the invention will become apparent from
15 the following description and claims, when taken with the accompanying drawings, wherein similar reference characters refer to similar elements throughout and in which:

FIGURE 1 is a block diagram of one embodiment of a data
compression and decompression system in accordance with the invention;

20 FIGURE 2 is a flowchart of operations that may be performed by a compression system implemented according to the invention;

FIGURE 3 is a flowchart of operations that may be performed by a decompression system implemented according to the invention;

25 FIGURE 4 is a block diagram of one embodiment of a computer configured to perform compression and/or decompression methods according to the invention;

FIGURE 5 is a block diagram of one embodiment of a data network system incorporating compression and decompression in accordance with the invention; and

30 FIGURE 6 is a block diagram of another embodiment of a data network system incorporating compression and decompression in accordance with the invention.

DESCRIPTION OF EXEMPLARY EMBODIMENTS

FIGURE 1 is a block diagram of a data compression system C (top half of FIGURE 1) and a data decompression system D (bottom half of FIGURE 1) in accordance with one embodiment of the invention. Briefly, a first processor 20 executes a data compression program 22 that uses synthetic history to compress input data. As represented by the dashed line 24, the compressed data is sent to a second processor 26. The second processor 26 executes a data decompression program 28 that uses synthetic history to decompress the compressed data.

The operation of the components of FIGURE 1 may be better understood by reference to FIGURES 2 and 3. FIGURE 2 is a flowchart of one embodiment of operations that may be performed by the data compression stage C. FIGURE 3 is a flowchart of one embodiment of operations that may be performed by the data decompression stage D.

The method of FIGURE 2 commences at block 100. As some point in time prior to the beginning of the compression process, synthetic history data is created for each data type that is to be compressed using synthetic history compression (block 102). The synthetic history data for the data types is stored in one or more synthetic history data files 30 in a data memory 32.

As discussed above, the data to be compressed (e.g., the "text" referred to in the Background section) may represent various information including, for example, conventional character text, image data, video data or audio data. In addition, much of this data may be classified as a particular data type; for example, language type (French, English, etc.) or document type (electronic mail, JAVA™, HTML documents, executable code, etc.).

In one embodiment of the invention, the synthetic history file 30 for a given data type contains a collection of characters and strings that are likely to appear in data of that type. The synthetic history data (e.g., the "typical data") for a given data type may be chosen using statistical methods. For example, character or string probabilities for a given type may be generated

by analyzing a large number of files of that type.

At the beginning of the compression process for a given set of data, the processor 20 (FIGURE 1) receives input data from an input data source 34 (block 104). Two types of input data are typical: static file data or streaming data. In the first case (described in more detail below in conjunction with FIGURE 4), the input data source 34 may be a conventional data memory such as a disk drive or random access memory (RAM). In the case of streaming data (described in more detail below in conjunction with FIGURES 5 and 6), the input data source 34 may be a data interface device that receives, for example, streaming packet data from a data network.

At block 106, a data type identifier routine 36 analyzes either the input data or information associated with the input data to ascertain the data type of the input data. For example, in the first case the data type identifier 36 may perform a relatively fast analysis of the input data. This may involve searching for strings that commonly appear at the beginning of particular documents. For example, an e-mail file may contain headers such as "From" and "To". A Microsoft™ Word™ document may contain a common signature such as "WordDocument". Alternatively, the identification process may involve searching for a word that is very likely to appear in one language, but not in other languages.

One example of the second case relates to compression of streaming packet data in an Internet environment. Here, the data type identifier 36 ascertains the TCP port to which the incoming packet data is associated by analyzing the header information in the packet. TCP ports are defined by the TCP/IP protocol that is used by many applications to route data over the Internet. In the instant case, the TCP port number may be used to identify a data type when it is known that a specific type of file typically originates from a particular TCP source port. As an example, the compression device may be configured to compress data originating from a particular server. That is, all traffic from the server may be routed through the compression device. In this case, it may be possible to predict the types of data that typically are sent by the server on a particular TCP port. From the above, it should be

understood that many other methods of data type identification may be used in practicing the invention.

After the data type has been identified, a data selector function 38 retrieves the associated "typical data" (i.e., the synthetic history data) from the data memory 32 (block 108). Next, the data compressor 22 compresses the "typical data" (block 110) and discards the resulting compressed data (block 112). In conjunction with these steps, the resulting history data (preliminary history data) is stored in a history file 40 that will be used during the compression of the input data. Thus, at this stage of the compression process, the history file 40 has been pre-loaded with history data representative of the "typical data" for the input data type.

The input data may now be compressed using the preliminary history (block 114). In one embodiment, the data compressor 22 uses a Lempel-Ziv compression method. It should be understood, however, that other methods of data compression such as adaptive Huffman coding may be used in practicing the invention.

At block 116, a reference is associated with the compressed data. This reference is used by the decompression stage D to determine which synthetic history to use when decompressing the compressed data. For example, the reference may identify the data type. In the Internet example discussed above, the reference may be inserted into the header of a packet within which the compressed data is transmitted.

At block 118, the processor 20 sends the compressed data to an output data destination 42 and the process ends at block 120. As above, the destination 42 for the output data may be, for example, a data file or a data stream. Thus, the output data destination 42 may comprise, for example, a data memory or a data interface device as described below in conjunction with FIGURES 4, 5 and 6.

In the embodiment described above, the output data includes only the compressed input data and, if applicable, the reference. The output data need not include the compressed "typical data" or the history data.

Referring to FIGURE 3, a decompression method is described.

beginning at block 150. At block 152, synthetic history data is created for each data type that is to be decompressed using a synthetic history. This step is performed in a similar manner as described above in conjunction with block 102, except that the "typical data" is pre-compressed using a compression algorithm that is compatible with the algorithm used by the data compressor 22 described above. Thus, in this embodiment, the decompressor's synthetic history data files 44 contain "typical compressed data."

At the beginning of the decompression process for a given set of data, the processor 26 (FIGURE 1) receives compressed input data from an input data source 46 (block 154). As stated above in conjunction with block 104, two types of input data are typical: file data or streaming data. Thus, the input data source 46 may comprise components similar to those discussed above.

At block 156, a data type identifier routine 48 analyzes the compressed input data or information associated with the input data to determine the data type of the input data. For example, the data type identifier 48 may determine the data type by reading a reference that was sent with the data as discussed above in conjunction with block 116.

After the data type has been identified, a data selector function 50 retrieves the associated "typical compressed data" (i.e., the synthetic history data) from the data memory 52 (block 158). Next, the data decompressor 28 decompresses the "typical compressed data" (block 160) and discards the resulting decompressed data (block 162). In conjunction with these steps, the resulting history data (preliminary history data) is stored in a history file 54 that will be used during the decompression of the input data. Thus, at this stage of the decompression process, the history file 54 has been pre-loaded with history data representative of the "typical compressed data" for the input data type.

At block 164, the data decompressor 28 uses the preliminary history to decompress the compressed input data. To this end, the data decompressor 28 uses a decompression algorithm that is complementary to the compression algorithm used by the data compressor 22.

At block 166, the processor 26 sends the decompressed data (which is a replica of the original data) to an output data destination 56 and the process ends at block 168. As above, the destination for the output data may be, for example, a data file or a data stream. Thus, the output data destination 56 may comprise components similar to those discussed above.

In another embodiment of the invention, the synthetic history data used by the data compressor 22 may comprise actual history data. For example, in this embodiment, the "typical data" is not stored in the synthetic history data file 30 as discussed above in conjunction with block 102 in FIGURE 2.

Instead, the history data that would result from the compression of the "typical data" is stored in the synthetic history data file 30. This history data may be generated, for example, by explicitly defining the history data or by actually compressing the "typical data". In the latter case, the history data is saved and the compressed "typical data" discarded.

When input data is to be compressed, the step of selecting the typical data (block 108) may simply involve copying the history data from the synthetic history data file (e.g., file 30) to the history file that is used for compressing the input data (e.g., file 40). In addition, the real-time steps of compressing the "typical data" (block 110) and discarding the compressed "typical data" (block 112) are omitted.

The decompression process may be modified in a similar manner. The synthetic history data used by the data decompressor 28 may comprise actual history data. For example, rather than storing the "typical compressed data" in the synthetic history data file 44 as discussed above in conjunction with block 152 in FIGURE 3, the history data that would result from the decompression of the "typical compressed data" may be stored in the synthetic history data file 44. This history data may be generated, for example, by explicitly defining the history data or by actually decompressing the "typical compressed data". In the latter case, the history data is saved and the decompressed "typical compressed data" discarded.

When input data is to be decompressed in this embodiment, the step of selecting the typical data (block 158) may simply involve copying the

history data from the synthetic history file (e.g., file 44) to the history file that is used to decompress the input data (e.g., file 54). In addition, the real-time steps of decompressing the "typical compressed data" (block 160) and discarding the decompressed "typical compressed data" (block 162) are omitted.

From the above, it should be understood that synthetic history data may include, for example, data that may be used to generate history data, actual history data that is the product of a compression or decompression operation, or predefined history data.

In another embodiment of the invention, synthetic history data (such as the compressed "typical text") may be sent from the data compressor stage C to the data decompressor stage D. In this example, the steps described in blocks 112, 116, 152, 156 and 158 may be omitted because the data decompressor 28 may simply use the compressed "typical text" (generated at block 110) at block 160. In practice, this technique may be less efficient than the previously described techniques in many applications. Nevertheless, it should be understood from the above that many adaptations involving the use of synthetic histories are possible in practicing the invention.

FIGURE 4 illustrates some of the components that may be incorporated into a device 200 that performs data compression and/or data decompression in accordance with the invention. A processor 202 executes program code (not shown) stored in a program memory 204 to perform, for example, the methods described herein in conjunction with FIGURES 1-3 and 5-6. Typically, the program memory 204 comprises a read only memory (ROM) device or a semi-permanent data memory such as a flash memory. The computer 200 also includes at least one storage memory 206 for storing dynamic data. Typically, the storage memory 206 comprises a random access memory (RAM) device or a disk drive.

The program code may be pre-loaded into the program memory 204, for example, at the factory. Alternatively, in embodiments that are connected to a data network such as the Internet, the program may be downloaded from a server via the data network.

In another embodiment, the program code may be stored on a removable media 208 such as a CD-ROM or a floppy disk. In this case the computer 200 would include a removable media drive 210 such as CD-ROM drive or a floppy disk drive. The program code may then be downloaded into the program memory 204 or, in some cases, accessed directly by the processor 202 from the removable media 208.

One or more data interfaces 212 may enable the computer 200 to send or receive data to or from external devices (not shown). This data may include the program data, the original data, the compressed data or the decompressed data. Examples of data interfaces 212 include serial or parallel ports, bus interfaces, or data network interfaces. The data network example is discussed in more detail below in conjunction with FIGURE 5 and 6.

The teachings of the invention may be used for file compression schemes that attempt to use disk drive space more efficiently by storing data in a compressed format on the system disk drive. Such a scheme may be used, for example, by a computer operating system and implemented in the embodiment of FIGURE 4 in the following manner. The computer 200 includes an operating system installed in the program memory 204 and executed by the processor 202. The operating system incorporates the synthetic history compression and decompression functions as treated herein.

Thus, the operating system may compress files before they are saved to the system hard disk drive (e.g., storage memory 206). Similarly, the operating system may decompress files after they are read from the system hard disk drive.

FIGURE 5 illustrates one embodiment of the invention that compresses streaming data on-the-fly. Specifically, the invention is employed in equipment installed in a path in a data network. In general, this equipment incorporates features and functional elements similar to those of the embodiments described above to accomplish data compression and decompression using synthetic histories.

Packet-based data networks (such as the Internet) transfer information

between computers and other equipment using a data transmission format known as packetized data. The stream of data from a data source (e.g., a host computer) is divided into variable or fixed length "chunks" of data (i.e., packets). Switches (e.g., routers) in the network route the packets from the source to the appropriate data destination. In many cases, the packets may be relayed through several routers before they reach their destination. Once the packets reach their destination, they are reassembled to regenerate the stream of data.

Conventional packet-based networks use a variety of protocols to control data transfer throughout a network. For example, the Internet Protocol ("IP") defines procedures for routing data through a network. To this end, IP specifies that the data is organized into frames each of which includes an IP header and the associated data. The routers in the network use the information in the IP header to forward the packet through the network. In the IP vernacular, each router-to-router (or switch-to-router, etc.) link is referred to as a hop.

In FIGURE 5, a router 220 at one end of a hop in the network sends packets to another router 222 at the other end of the hop. Some of the packets sent over the hop may be associated with data types that can be compressed using pre-defined synthetic histories (not shown). In accordance with the invention, a compressor 224 compresses the data in these packets using the synthetic histories. On the other end of the path, a decompressor 226 decompresses the data in the compressed packets using pre-defined synthetic histories.

In practice, the link between the routers 220 and 222 may be either a permanent or temporary link. The link may be used to transfer unmodified layer 3 protocol packets. Layer 3 is a network layer protocol and encompasses, for example, the Internet Protocol ("IP") and those that conform to the OSI ("Open System Interconnection") reference model.

The compressor 224 processes an inbound stream of packets from the router 220. One or more network interfaces 228 in the compressor terminates the packet protocols and provides the packet data to a processor

230. When the devices (i.e., the compressor 224 and the decompressor 226) are installed between the routers 220 and 222 as illustrated in FIGURE 1, the network interface 228 connects to a wide area network ("WAN") as described above. In some embodiments, the compressor 224 may be installed farther
5 up the link (i.e., before the router 220). In this case, the network interface 228 may connect to a local area network ("LAN"). The network interface 228 in the latter type of system will include a LAN-type interface such as an Ethernet interface.

The processor 230 performs data compression and other processes
10 such as those as described above in conjunction with FIGURES 1, 2 and 4. To reduce the complexity of FIGURE 5, the data memories and other components associated with the processor 230 are not depicted.

After the processor 230 compresses the input data, the processor 230 sends the packets to the network interface 228. The network interface 228
15 processes the packet data and provides the appropriate physical and data link layers to interface to the network (as represented by line 232). In practice, separate input and output network interface components may be used in the compressor 224. The details of the operation and implementation of the network interfaces 224 as described are well known in the IP data
20 networking art. Accordingly, these aspects of the compressor 224 will not be treated in detail here.

As represented by the line 232 in FIGURE 5, packets from the compressor 224 are routed over the network to the decompressor 226 on the other end of the path. A network interface 234 terminates the physical and
25 data link layers and provides network layer (IP) packets to a processor 236. The details of the operation and implementation of the network interface 234 may be similar to those aspects of the network interfaces (e.g., interface 228) discussed above.

The processor 236 in the decompressor 226 performs data
30 decompression and other processes such as those as described above in conjunction with FIGURES 1, 3 and 4. To reduce the complexity of FIGURE 5, the data memories and other components associated with the processor

236 are not depicted.

After the processor 236 decompresses the input data, the processor 236 sends the packets to the network interface 234. The network interface 234 processes the packet data and provides the appropriate physical and data link layers to interface to the network. The data is then forwarded to the router 222.

FIGURE 6 illustrates an embodiment in which the compression and decompression methods of the invention are integrated as software modules in devices 240 that are installed at each end of a predefined path in a network. The devices 240 may be routers, bridges, switches, modems or any other device in the network that handles packet traffic.

The packet compression and decompression operations performed by the embodiment of FIGURE 6 are similar to those described above in conjunction with FIGURES 2-5. Compression software modules 242 and decompression software modules 244 are linked to software modules 246 in the devices in a manner that enables the compression software modules 242 and the decompression software modules 244 to intercept and process packets routed through the devices 240. To reduce the complexity of FIGURE 6, other components in the devices 240 such as data memories and network interfaces are not depicted.

Typically, the compression and decompression software modules 242 and 244 may be implemented along the transmission path in a device 240 where the packets are fully visible. For example, some of the packets flowing through the network may be encrypted. Thus, the compressor and decompressor software modules 242 and 244 may be linked in to the device modules 246 so that the compression and decompression software modules 242 and 244 have access to decrypted data.

In FIGURE 6, the compression and decompression modules 242 and 246 are installed on both sides of a duplex link. Accordingly, packet traffic traveling in either direction on the link may be compressed according to the invention.

FIGURE 6 also illustrates that the invention may be used on more than

a single IP hop. In FIGURE 6, the packets are routed through a network 248 (e.g., the Internet) and, as a result, they may be routed over several hops. In this case, appropriate routing provisions should be made to ensure that all compressed packets are routed to the same receive module at the other end of the path. This may include, for example, defining static routes using IP tunneling.

In the compression scheme above, it is important to maintain the reliability of the link when multiple packets are to be compressed or decompressed using the same history file. This is because, in order to decompress packet "n," the decompression module 244 must first decompress packets "1" through "n-1." Reliability may be provided by the reliability mechanism associated with TCP, HDLC (in its reliable mode) or PPP (in its reliable mode).

In many of the embodiments described above, various initialization procedures may be performed. For example, all history files may be erased and various compression parameters may be exchanged between a paired compressor and decompressor. In the data networking embodiments, these initialization procedures may be accomplished using a relatively simple three-way handshake such as the one used in TCP.

From the above, it may be seen that the invention provides an improved method of compressing data and increasing data throughput in a network. In many applications, a system or method constructed or implemented according to the invention will find history data that matches the data being compressed earlier in the compression process than conventional compression methods. As a result, data can be compressed more quickly and to a higher degree of compression. In particular, a system or method constructed or implemented according to the invention may provide significantly higher compression ratios for relatively small data files (e.g., files smaller than ten kilobytes).

While certain specific embodiments of the invention are disclosed as typical, the invention is not limited to these particular forms, but rather is applicable broadly to all such variations as fall within the scope of the

appended claims. To those skilled in the art to which the invention pertains many modifications and adaptations will occur.

For example, the devices may be installed at various locations within the network. The invention may be implemented using a variety of hardware and software architectures. The teachings of the invention are applicable to numerous compression algorithms and compression history techniques in addition to those described above. The system and methods of the invention may be used to compress and decompress various types of data. Also, many techniques for identifying data types may be used. Thus, the specific structures and methods discussed in detail above are merely illustrative of a few specific embodiments of the invention.

WHAT IS CLAIMED IS:

1. A method of compressing data comprising the steps of:
generating synthetic history data associated with at least one
5 data type;
receiving data to be compressed;
determining a data type of the received data;
selecting synthetic history data associated with the determined
data type; and
10 compressing the received data using the selected synthetic
history data.
2. The method of claim 1 wherein the generating step is performed prior
to the compressing step.
- 15 3. The method of claim 1 wherein the determining step includes analyzing
information associated with the received data.
4. The method of claim 3 wherein the information identifies a TCP port.
- 20 5. The method of claim 1 wherein the determining step includes analyzing
the received data.
6. The method of claim 1 wherein the synthetic history data includes
25 information frequently present in data of a given data type.
7. The method of claim 6 wherein the generating step includes
compressing the synthetic history data.
- 30 8. The method of claim 1 wherein the generating step comprises defining
history data.

9. The method of claim 1 wherein the compressing step comprises using a Lempel-Ziv algorithm to compress the received data.
- 5 10. The method of claim 1 wherein the compressing step comprises using an adaptive Huffman algorithm to compress the received data.
11. The method of claim 1 further comprising the step of associating a reference with compressed received data, wherein the reference is used to
- 10 select synthetic history data for decompressing the compressed received data.
12. The method of claim 11 wherein the reference identifies a data type.
- 15 13. A method of compressing data comprising the steps of:
receiving data to be compressed;
prior to compressing the received data, generating synthetic history data that is associated with the received data; and
compressing the received data using the synthetic history data.
- 20 14. The method of claim 13 wherein the synthetic history data is associated with at least one data type.
15. The method of claim 14 further comprising the step of determining a
- 25 data type of the received data.
16. The method of claim 15 further comprising the step of selecting the synthetic history data based on the determined data type.
- 30 17. The method of claim 15 wherein the determining step includes analyzing information associated with the received data.

18. The method of claim 17 wherein the information comprises a TCP port.

19. The method of claim 15 wherein the determining step includes analyzing the received data.

20. The method of claim 13 wherein the synthetic history data includes information frequently present in data of a given data type.

21. The method of claim 20 wherein the generating step includes compressing the synthetic history data.

22. The method of claim 13 wherein the generating step includes defining history data.

23. The method of claim 13 wherein the compressing step comprises using a Lempel-Ziv algorithm to compress the received data.

24. The method of claim 13 wherein the compressing step comprises using an adaptive Huffman algorithm to compress the received data.

25. The method of claim 13 further comprising the step of associating a reference with compressed received data, wherein the reference is used to select synthetic history data for decompressing the compressed received data.

26. The method of claim 25 wherein the reference is an identifier associated with a data type.

27. A method of decompressing data comprising the steps of:

- generating synthetic history data associated with at least one data type;
- receiving data to be decompressed;

determining a data type of the received data;
selecting synthetic history data associated with the determined
data type; and
decompressing the received data using the selected synthetic
5 history data.

28. The method of claim 27 wherein the determining step comprises
analyzing a reference associated with the received data.

10 29. A method of decompressing data comprising the steps of:
receiving data to be decompressed;
prior to decompressing the received data, generating synthetic
history data that is associated with the received data; and
15 decompressing the received data using the synthetic history
data.

30. The method of claim 29 wherein the synthetic history data is
associated with at least one data type.

20 31. The method of claim 30 further comprising the step of determining a
data type of the received data.

32. The method of claim 31 further comprising the step of selecting the
synthetic history data based on the determined data type.

25 33. The method of claim 31 wherein the determining step comprises
analyzing a reference associated with the received data.

30 34. A data compression system comprising:
a data memory for storing synthetic history data associated with
at least one data type;
a data type identifier for determining a data type of received

data;
 a data selector for selecting synthetic history data associated
 with the determined data type; and
 a compressor for compressing the received data using the
 selected synthetic history data.

35. The data compression system of claim 34 further comprising at least
 one data network interface.

36. A data decompressing system comprising:
 a data memory for storing synthetic history data associated with
 at least one data type;
 a data type identifier for determining a data type of compressed
 data;
 a data selector for selecting synthetic history data associated
 with the determined data type; and
 a data decompressor for decompressing the compressed data
 using the selected synthetic history data.

37. The data decompression system of claim 36 further comprising at least
 one data network interface.

38. A computer program product comprising:
 a computer usable medium having computer readable program
 code means embodied therein for compressing received data, the
 computer readable program code means in said computer program
 comprising:
 storage medium in which synthetic history data
 associated with at least one data type is stored;
 means for determining a data type of the received data;
 means for selecting synthetic history data associated with
 the determined data type; and

means for compressing the received data using the
selected synthetic history data.

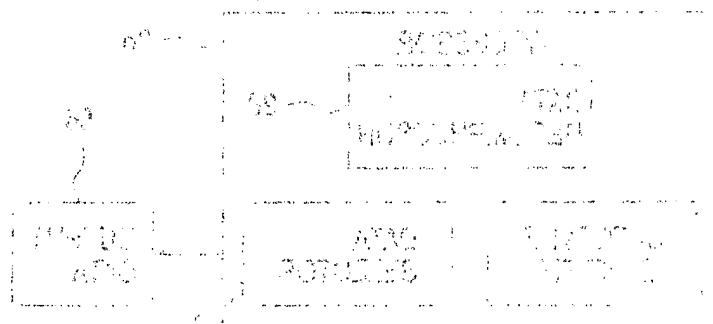
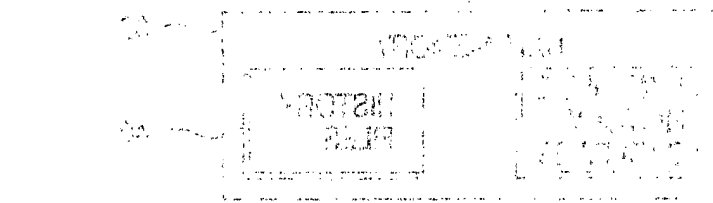
39. A memory for storing data, said memory having a data structure stored
therein, said data structure including the stored data and comprising:

storage medium in which synthetic history data
associated with at least one data type is stored;

means for determining a data type of received data;

means for selecting synthetic history data associated with
the determined data type; and

means for compressing the received data using the
selected synthetic history data.



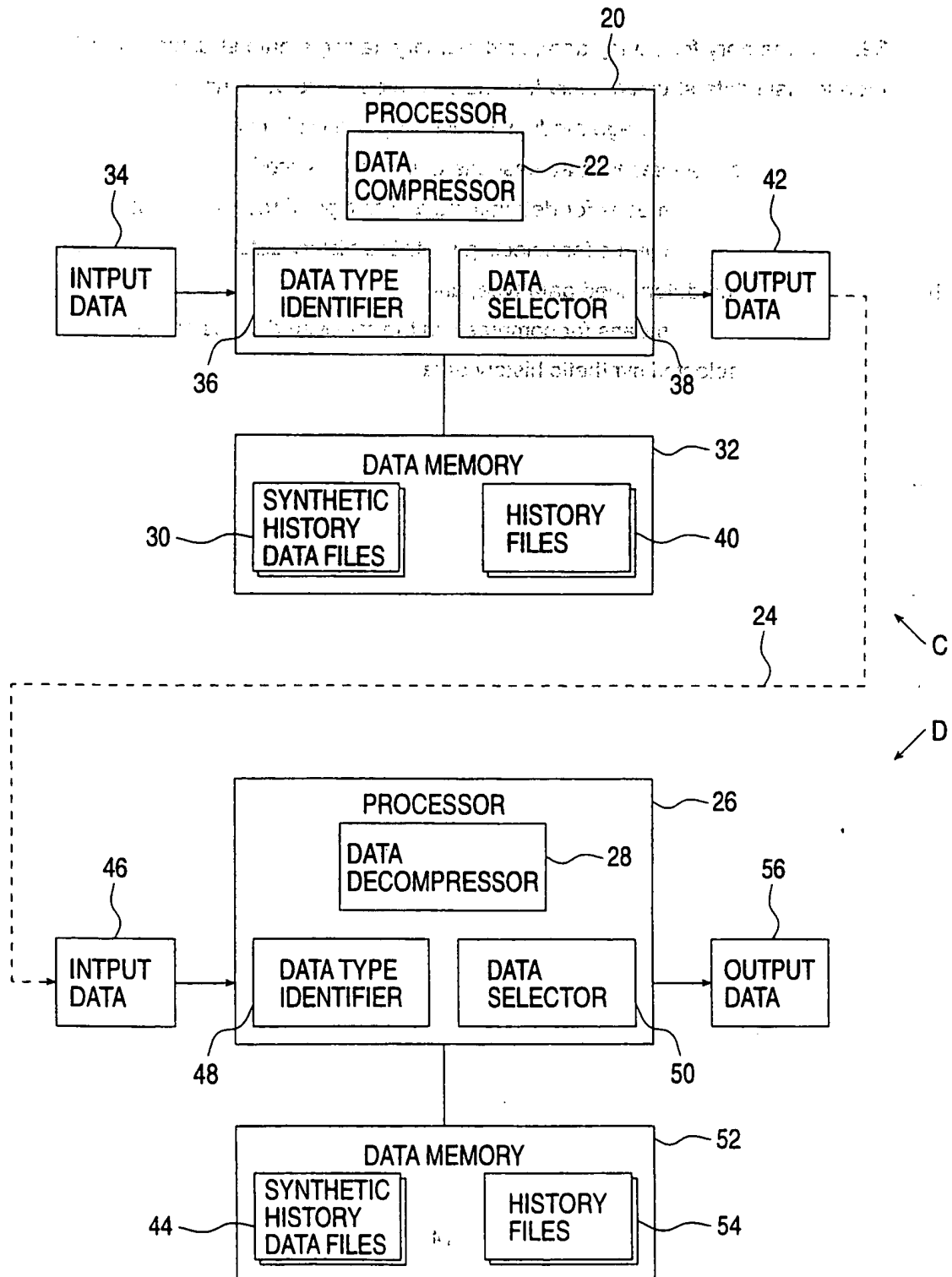


FIG.1

2/4

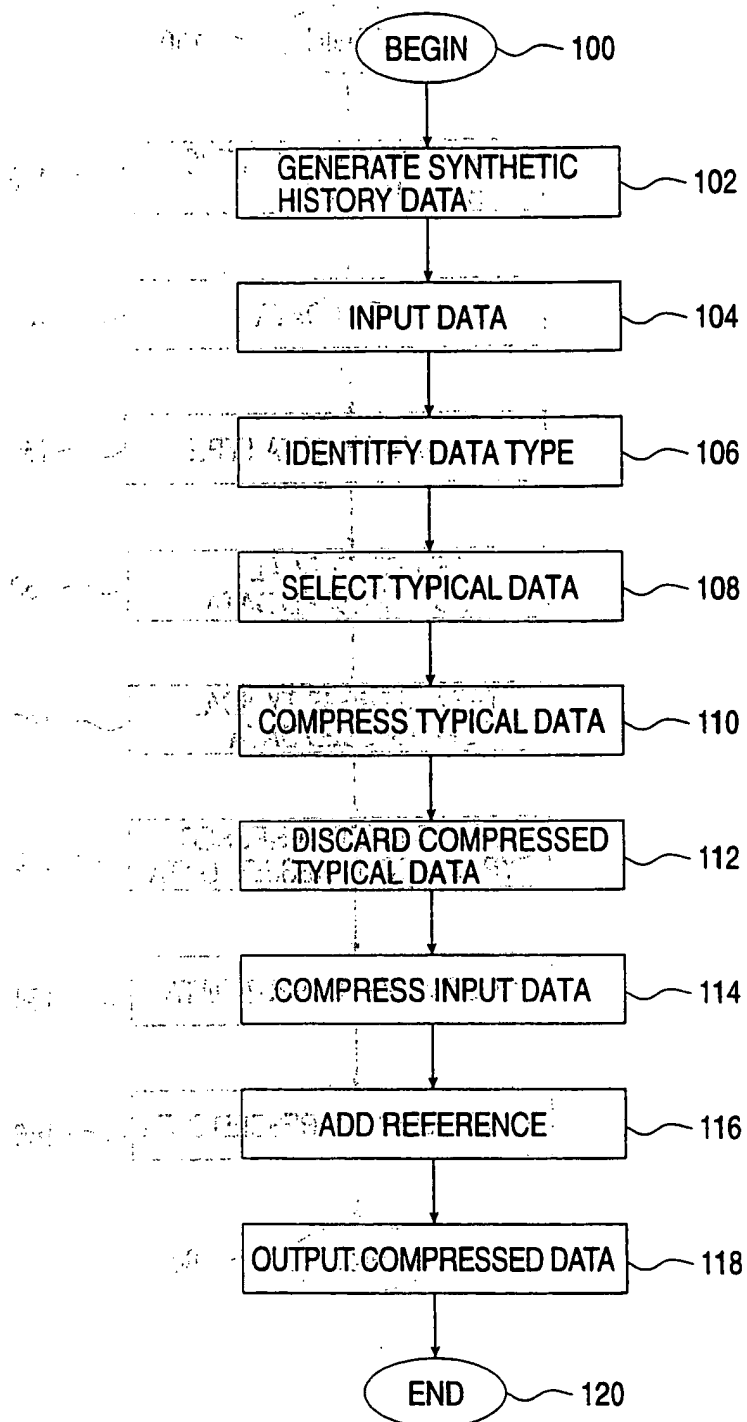


FIG. 2

3/4

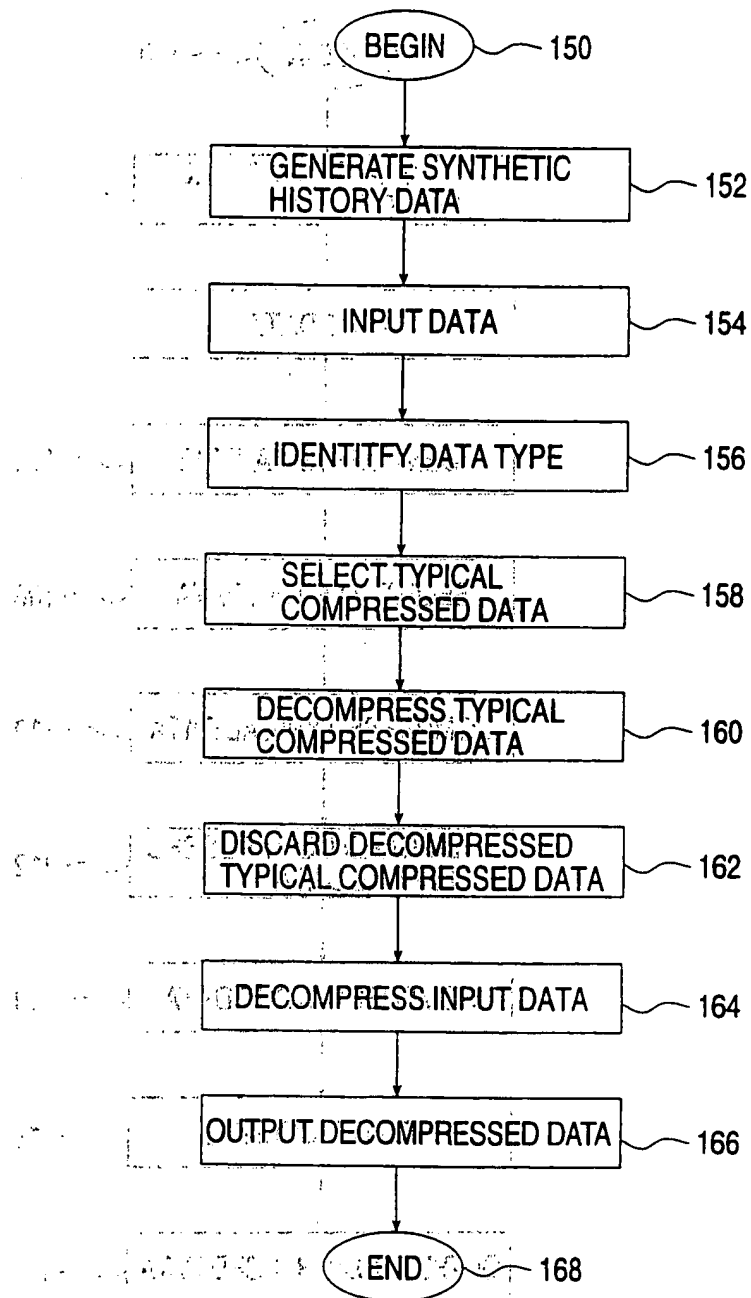


FIG.3

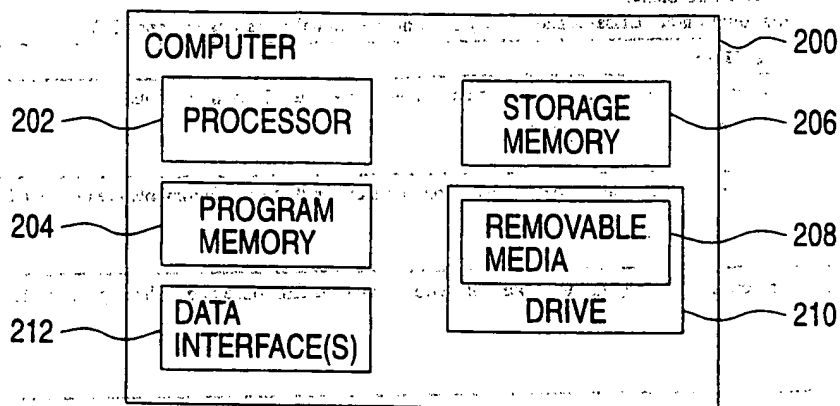


FIG. 4

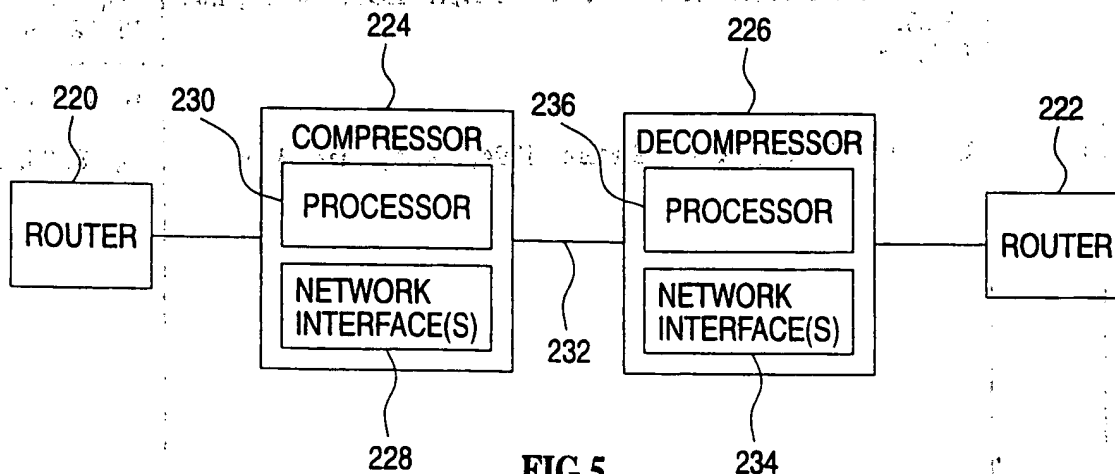


FIG. 5

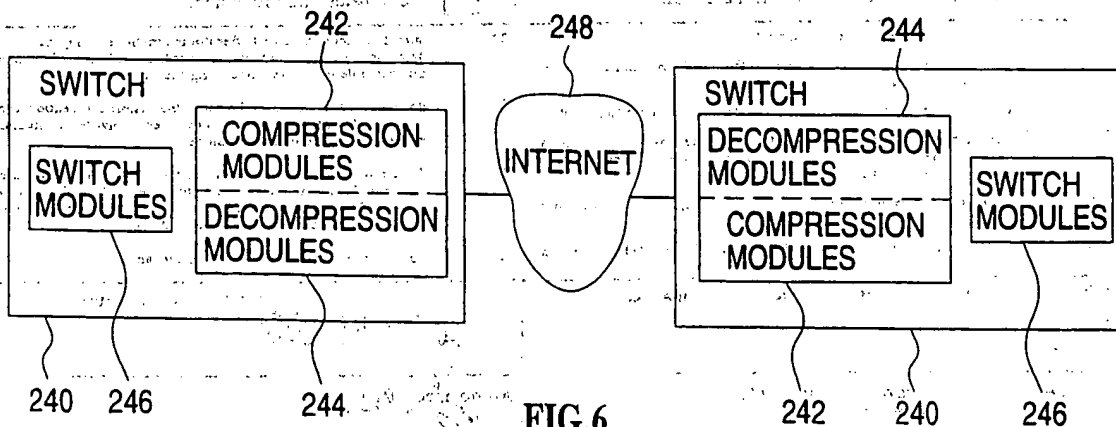


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/29944

A. CLASSIFICATION OF SUBJECT MATTER		
IPC () : G06K 9/00 US CL : Please See Extra Sheet. According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) U.S. : 382/232, 239, 247, 248, 249, 250		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X ---,P Y	US 5,896,403 A (NAGASAKI et al) 20 April 1999, col. 64, lines 45-60. FIG. 15	1-3, 5-17, 19-34, 36, 38, 39 ----- 4, 18, 35, 37
Y	US 5,293,379 A (CARR) 08 March 1994, col. 5, lines 33-36.	4, 18, 35, 37
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents: *A* document defining the general state of the art which is not considered to be of particular relevance *E* earlier document published on or after the international filing date *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) *O* document referring to an oral disclosure, use, exhibition or other means *P* document published prior to the international filing date but later than the priority date claimed *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents; such combination being obvious to a person skilled in the art *Z* document member of the same patent family		
Date of the actual completion of the international search 23 MARCH 2000		International search report 28 APR 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230		Authorized officer ALI BAYAT Telephone No. (703) 306-5915

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/29944

A. CLASSIFICATION OF SUBJECT MATTER:

US CL:

382/232, 239, 247, 248, 249, 250

[illegible]

W. J. G. S.

POSTAGE WILL BE PAID BY ADDRESSEE

1. The first step is to identify the problem or question that needs to be answered. This involves understanding the context and the specific requirements of the task.

Yucca elata L.

[illegible]

Figure 1. The effect of the concentration of the *Agrobacterium* suspension on the transformation efficiency of *Agrobacterium* strains. The *Agrobacterium* strains were grown in the YEA medium for 24 h at 28°C. The cell concentration of the *Agrobacterium* strains was adjusted to 1.0 × 10⁸ cells/ml. The cell suspension was mixed with the plant tissue and incubated for 24 h at 28°C. The plant tissue was then cultured on the selective medium. The transformation efficiency was determined by the number of transformants per plant tissue. The data are the mean ± SD of three independent experiments.

[illegible]

1. The first step in the process of the investigation is the identification of the problem. This is done by the investigator who is responsible for the study. The investigator must first identify the problem that is being investigated. This is done by the investigator who is responsible for the study. The investigator must first identify the problem that is being investigated.